REMARKS

This Supplemental Amendment Under 37 CFR 1.116 is submitted to supplement the Amendment Under 37 CFR 1.116 filed concurrently herewith, and to respond to the outstanding final Office Action dated December 4, 2007, in this application. The concurrently filed Amendment Under 37 CFR 1.116 is submitted to present applicants' claims, as they were to be submitted in the October 12, 2007, Amendment Under 37 CFR 1.111. The October 12, 2007, Amendment was responsive to the July 13, 2007, Office Action, but was inadvertently submitted without the "strikethrough deletions" in the amended claims as presented. Hence, the Amendment Under 37 CFR 1.116 includes the strikethrough deletions that were inadvertently left out of the Listing of Claims in the October 12, 2007, Amendment, and the underlined additions, as submitted in the October 12, 2007, Amendment.

Consequently, the Listing of Claims in the instant Supplemental Amendment Under 37 CFR 1.116 reflects the state of the claims as amended by the Amendment Under 37 CFR 1.116 (filed concurrently herewith).

Claims 6, 9, 17, 21 and 24 are amended hereby to address the rejections under 35 USC §112, second paragraph, set forth at paragraph 7 of the outstanding final Office action (dated December 4, 2007). Claims 10 and 14 are also amended to address the rejection of claims 10-17 under the second paragraph of section 112, set forth at paragraph 8 of the outstanding formal Office Action. The amendments are formal, and do not change the scope of the amended claims; no new matter is added. Claims 1-24 remain pending hereinafter, where claims 1, 10 and 18 are the independent claims.

Response To Claim Objections and Compliance with 37 CFR 1.121

Applicants respectfully assert that the outstanding claim objections are addressed in the concurrently filed Amendment Under 37 CFR 1.116, which obviates the stated objections. Applicants, therefore, respectfully request withdrawal of the claim objections.

Provisional Rejection of claims 6, 14 and 21 Under Non-Statutory Obviousness-Type Double Patenting Doctrine

In the December 4, 2007 Office Action, the Examiner rejects dependent claims 6, 14 and 21 under the judicially created doctrine of obviousness-type double patenting, over claims 1, 18 of co-pending, commonly owned application Serial No. 11/227,761 (the '761 application").

With respect to dependent claim 6, the Examiner states that while independent claim 1 of the '761 application is not identical, dependent claim 6 recites dynamically updating an operating system by identifying references to said first code component and replacing the identified references to said first code component to said new code component. The Examiner asserts that cited independent claim 1 of the '761 application does not explicitly recite separating the first code component into objects grouped in table, but that one skilled in the art would infer that the reference pointer could be changed to point to a new object instead of a factory configured object, and removing the factory object. The Examiner concludes that providing said runtime (new) factory object in a form of runtime table entry (as in a table storing reference entries) so that the table-represented reference pointers identify during the dynamic replacement of reference objects, in view of well-known approach using reference table to interrelate dynamic code referencing to provide program resolution support at runtime.

With respect to dependent claims 6, 14 and 21, the Examiner indicates that independent claim 18 of the '761 application recites updating an operating system without rebooting, identifying references to said first code component and replacing the identified references to said first code component to said new code component, but that the '761 application does not explicitly recite by what is described by the Examiner as separating the first code component into objects grouped in table, whereby references to said objects are entered in the table, but that in view of the structure to store the definition referring to the object instance being swapped from above, this reference table limitation would have been obvious.

Applicants respectfully disagree. A non-statutory obviousness-type double patenting rejection is appropriate where the conflicting claims are either anticipated by, or would have been obvious over the referenced claims. MPEP 804; In re Berg, 140 F.3d 1428, 46 USPQ2d

1226 (Fed. Cir. 1998). Such rejections are in order where the claimed subject matter is not patentable distinct from the cited claimed subject matter.

Applicants' dependent claims 6, 14 and 21 are not patentably indistinct from claims 1 and 18 of the '761 application. That is, dependent claims 6, 14 and 21 are patentably distinct from the subject matter of independent claims 1 and 18 of the '761 application because the subject matter of applicants' instant independent claims 1, 10 and 18, from which dependent claims 6, 14 and 21 depend, is patentably distinct from claims 1 and 18 of the '761 application. The inventions set forth in applicants' dependent claims 6, 14 and 21 include the inventive subject matter and claimed limitations of independent claims 1, 10 and 18, respectively, as modified by the further limitations set forth by the language of respective dependent claims 6, 14 and 21.

Dependent claims 6, 14 and 21, are not anticipated by, or an obvious variation of independent claims of 1 and 18 of the '761 application. Analyses employed in an obviousness-type double patenting rejection parallel the guidelines for analysis under 35 USC §103. MPEP 804; In re Braat, 937 F.2d 589 (fed. Cir. 1981). Where an independent claim is patentable in view of a reference under section 102 and 103, claims depending from the patentably distinct independent claim are also patentably distinct under US Law. MPEP 608.01(n) (II) states that a proper dependent claim shall not conceivably be infringed by anything that would not also infringe the base (independent) claim from which the dependent claim depends. MPEP 608.01(n)(III) states that "[e]xaminers are reminded that a dependent claim is directed to a combination including everything recited in the base claim and what is recited in the independent claim. It is this combination that must be compared with the prior art, exactly as if it were presented as one independent claim."

Applicants respectfully assert, therefore, that the Examiner has failed to establish a prima facie case of obviousness under the judicially created doctrine of non-statutory double patenting. That is, because the subject matter of independent claims 1, 10 and 18 is patentably distinguishable from the subject matter of claims 1 and 18 of the '761 invention, the subject matter of dependent claims 6, 14 and 21 that derive from those independent claims is patentably distinguishable from claims 1 and 18 of the '761 application for at least the reasons that

distinguish independent claims 1, 10 and 18 from claims 1, and 18 of the '761 application. Applicants, therefore, respectfully assert that dependent claims 6, 14 and 21 are not unpatentable in view of independent claims 1 and 18 of the '761 application under the judicially created doctrine of obviousness-type double patenting, and request withdrawal of the rejection of claims 6, 14 and 21 thereunder.

Response To Rejections Under 35 USC §102

Claims 1-24 were rejected under 35 USC §102(b) as unpatentable over US Patent No. 6,219,690 to Slingwine. The Examiner asserts with respect to independent claim 1 that Slingwine discloses in a computer system using an operating system to provide access to hardware resources, wherein said operating system provides access to said resources via a first source code component, a method of replacing said first source code component with a new source code component while said operating system remains active and while said operating system provides continual availability to applications of the hardware resources (Fig. 2, kernel running, col. 10, lines 54-67). The Examiner asserts that Slingwine further includes identifying references to said first source code component (108, col. 6, lines 41-50, col. 9, line 45 to col. 10, line 17, Fig. 4, Fig. 6); and replacing the identified references to said first source code with references to said new source code component (110, Fig. 3, 90, Fig. 3, Fig. 4).

With respect to independent claims 10, the Examiner asserts that Slingwine discloses a system for swapping source code in a computer system including an operating system, said operating system including at least one code component and providing continual availability of hardware resources by applications operational in the computer system (as stated in the claim 1 rejection), the system comprising:

means for identifying, while said operating system is active and providing continual access to said resources, references to a first code component of the operating system; and

means for replacing the identified references, while said operating system is active and providing continual access to said resources (interrupt; col. 18, lines 42-55; kernel 36; Fig. 2), to said first source code with references to a new source code component for the operating system;

all of which steps of identifying and replacing having been addresses in claim 1.

With respect to independent claim 18, the Examiner asserts all of which are addressed in the rejection of claim 1.

In response, applicants respectfully assert that independent claims 1, 10 and 18 are patentable distinguishable from Slingwine for at least the following reasons.

Applicants' invention is directed to a computer system that uses an operating system to provide access to hardware resources via a first code component and replacing the first code component with a new code component while the computer operating system remains active and while that operating system provides continual availability of the hardware resources to system applications. The method comprises the steps of identifying references to the first code component, and replacing the identified references to the first code component with references to the new code component, and swapping the first with the new code component. The swapping (hot swapping) is seamless to the hardware resources.

The invention in its various claimed embodiments perform a hot-swap, including that at establishing a *quiescent state* in the first code component, and identifying references to the first code component at the quiescent state, transferring the references to the first code component to the new code component, and then hot swapping the new code component safely and efficiently. The identified references are all of the outstanding references held by the clients to the first code component so that these references now point to the new code component. Efficiently achieving the operational state of the first code component (at quiescence) in the new code component is accomplished by tracking when threads associated with the first code component are created and destroyed. Quiescent state transfer of the references occurs via a best-negotiated protocol implemented by the invention between the first and new code components. Finally, using an object translation table, all calls to the first code component are rerouted to the new code component, in accordance with the invention.

Slingwine, as distinguished, discloses a substantially zero overhead mutual-exclusion apparatus and method that allow concurrent reading and updating data while maintaining data coherency. The data reading process executes the same sequence of instructions that would be

executed if the data were never updated, but rather than depending exclusively on overhead imposing locks, the mutual-exclusion mechanism tracks a thread execution history to determine safe times for processing a current generation of data updates while the next generation of data updates is concurrently being saved. When the last thread passes a quiescent state, a summary of thread activity triggers a callback processor that it is safe to end the current generation of updates. The callback processor restarts the summary of thread activity and initiates a next generation of updates. All data-updating threads pass through a quiescent state between the time they attempt to update data and the time the data are actually updated.

Slingwine does not swap a first code component with a new code component as does applicants' invention as set forth in independent claims 1, 10 and 18.

While the Examiner asserts that Slingwine discloses in a computer system using an operating system to provide access to hardware resources, wherein said operating system provides access to said resources via a first source code component, a method of replacing said first source code component with a new source code component while said operating system remains active and while said operating system provides continual availability to applications of the hardware resources (Fig. 2, kernel running, col. 10, lines 54-67), applicants respectfully disagree.

That is, Slingwine defines a quiescent state for a thread as that time that the thread is not accessing their mutual-exclusion mechanism. The mutual-exclusion mechanism is shown in Fig. 4, and described at col. 10, lines 52-67. Slingwine's mutual-exclusion mechanism is not equivalent to applicants' claimed method (mechanism), and is not understood, at the cited col. 10 text, to describe, in a computer system using an operating system to provide access to hardware resources, wherein said operating system provides access to said hardware resources via a first code component, a method of replacing said first code component with a new code component while said operating system remains active and while said operating system provides continual availability to the hardware resources by applications operational in the computer system.

While the Examiner asserts that Slingwine further includes identifying references to said first source code component (108, col. 6, lines 41-50, col. 9, line 45 to col. 10, line 17, Fig. 4, Fig. 6), applicants again respectfully disagree. The text at col. 6, lines 41-50, describes Slingwine's callback processor (104), which causes a next generation of Slingwine callbacks to become a current generation of callbacks. It does not show applicants step of identifying references to the first code component, required by each of applicants' independent claims. The text at col. 9, line 45 through col. 10, line 17, summarizes Slingwine's summary of thread activity, its generation counter, its thread counter, and its callback processor that uses each of the thread bits, generation counter, thread counter and callback processor. The text does not show applicants step of identifying references to the first code component, required by each of applicants' independent claims.

While the Examiner asserts that Slingwine further includes replacing the identified references to said first source code with references to said new source code component (110, Fig. 3, 90, Fig. 3, Fig. 4), applicants again respectfully disagree. Element 110 is a next generation and 90 is Slingwines' mutual exclusion mechanism. Neither can be said to applicants' replacing the identified references to the first code component with references to said new code component.

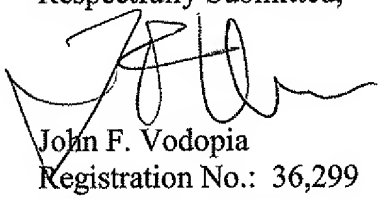
In view of the stated differences between amended independent claims 1, 10 and 18, taken as a whole, and Slingwine, applicants respectfully assert that independent claims 1, 10 and 18 are not anticipated by Slingwine under 35 USC §102(b), and respectfully request withdrawal of the rejections thereunder. Claims 2-9 depend from claim 1 and are patentable therewith; claims 11-17 depend from claim 10 and are patentable therewith; and claims 19-24 depend from claim 18 and patentable therewith. Applicant therefore respectfully asserts that claims 2-9, 11-17 and 19-24 are patentable in view of Slingwine Under Section 102 (b), and request withdrawal of same claim rejections in view of Slingwine.

The other references of record have been reviewed, and these other references, whether considered individually or in combination, also to not disclose or suggest the use of this invention as set forth in claims 1-24. Applicants, therefore, respectfully assert that claims 1, 4, 6,

7, 9, 10, 12-14, 16-17, 18, 19, 21, 23 and 24 fully comply with 37 CFR 1.121, claims 6, 9-17, 21 and 24, fully comply with 35 USC §112, second paragraph, claims 6,14 and 21 are not obvious under the judicially created doctrine of obviousness-type double patenting (over claims 1 and 18 of the '761 application) and claims 1-24 are not anticipated by Slingwine under 35 USC §102(b).

If the Examiner believes that a telephone conference with applicants' attorneys would be advantageous to the disposition of this case, the Examiner is asked to telephone the undersigned.

Respectfully Submitted,



John F. Vodopia
Registration No.: 36,299
Attorney for Applicants

Scully, Scott, Murphy & Presser. P.C.
400 Garden City Plaza, Suite 300
Garden City, New York 11530
(516) 742-4343

JFV:tb